Hogyan tanuljunk kevés információból is? A RIP-algoritmus továbbfejlesztett változatai

Biró Tamás

Amszterdami Egyetem (UvA)
Spuistraat 210, Amszterdam, Hollandia, e-mail: birot@nytud.hu

Kivonat A nyelvtanuló gyakran nem fér hozzá olyan információhoz, amely a nyelvészeti elméletekben központi szerepet játszik. Ez az információhiány a számítógépes szimulációk szerint hátráltathatja a nyelvelsajátítást. Kutatásom során az OT tanulóalgoritmusok sikerességét javítom Prince és Smolensky RIP-eljárásának továbbfejlesztésével.[1]

Kulcsszavak: Optimalitáselmélet (OT), Robust Interpretive Parsing, szimulált hőkezelés/lehűtés, genetikai algoritmusok, tanulóalgoritmusok.

1. Bevezetés: hiányzó információ a tanulás során

Vajon a *John loves Mary* mondat egy SVO vagy egy OVS nyelvből származik? Helyezzük magunkat a nyelvtanuló helyébe, aki hallja ezt a nyelvi adatot, és megfelelő ismerettel is rendelkezik a világról (vagyis tud a két személy közötti kölcsönös szerelemről): vajon milyen következtetést vonjon le a nyelvtanuló az elsajátítandó célnyelv szórendjére vonatkozóan? Amennyiben ezen a ponton (helytelenül) tárgy-ige-alany szórendet feltételez, akkor ez a nyelvi adat megerősítheti a nyelvtanulót téves hipotézisében, és a tanulási folyamat félrecsúszhat. Ha azonban egy más, óvatosabb algoritmust követ, és számol azzal, hogy jelenlegi hipotézise akár hibás is lehet, miközben a nyelvi adat több módon interpretálható, akkor a tanulás sikerrel járhat – mint azt rövidesen bemutatom.

A mondattanban az alany és a tárgy megkülönböztetése központi szerepet játszik, de az angol nyelvet éppen elsajátító nyelvtanuló számára nem hozzáférhető információ az, hogy az informáns (tanító) mely főnévi csoportot szánta alanynak, és melyiket tárgynak. A nyelvtan számos más pontján is hasonló problémák merülnek fel. Tizenegy hónapos kislányom megsimogatott a [*Mutasd meg, hol van*] *apa szeme!* utasításra, mert még nem sajátította el a [s]∼[š], valamint az [e]∼[i] közötti fonológiai különbségeket. Ezért a *szeme*∼*simi* párt szabad alternációként, nem pedig minimálpárként értelmezte. Apaként bízom benne, hogy kislányom esetében ez az egyszeri eset nem tereli vakvágányra a magyar fonológia elsajátítását.

[1] A szerző köszönetét fejezi ki a *Holland Tudományos Kutatási Alapnak* (NWO), amely a 275-89-004 számú Veni-projekt keretében az ismertetett kutatást támogatta.

Számítógépes nyelvészként célom a meglévő tanulóalgoritmusok továbbfejlesztése ugyanezen problémák elkerülése végett. Kutatásom tárgya az egyik leggazdagabb tanulhatósági irodalommal rendelkező kortárs nyelvészeti keret, az *Optimalitáselmélet* (OT) [1]. Az előbbiekben bemutatott problémára az OT hagyományos megoldása a *Robusztus Interpretatív Parszolás* (RIP) [2], amelyet a 3. fejezetben tárgyalok. A RIP teljesítménye azonban kívánnivalót hagy maga után. Ezért a 4. fejezetben két alternatívát mutatok be, amelyek teljesítményét az 5. fejezetben tesztelem.

Az első javaslat [3] a szimulált hőkezelés technikájából merít, és Boltzmann-eloszlást vezet be a megfigyelt nyelvi adat lehetséges interpretációin. A második javaslatot [4] a genetikai algoritmusok ihlették: párhuzamosan több, független tanulóalgoritmus fut, amelyek közösen interpretálják a bejövő nyelvi adatokat. Mielőtt azonban ezekre rátérnénk, foglaljuk össze az OT-val és tanulóalgoritmusaival kapcsolatos tudnivalókat.

2. Az optimalitáselmélet és tanulóalgoritmusai

Az *optimalitáselmélet* (*Optimality Theory*, OT) [1] alapgondolata az, hogy egy u bemenet (például mögöttes reprezentáció) arra a kimenetre (felszíni reprezentációra) képeződik le, amely optimalizál egy célfüggvényt. A gondolat önmagában nem új, hiszen számos tudományterület a fizikától a közgazdaságtanig – közöttük sok számítógépes kognitív modell is – célfüggvények optimalizációjával magyarázza jelenségeit. A nyelvészetben is gyakran hivatkozunk a „minél jobb" alakra. A nyolcvanas években a generatív nyelvészetben (különösen a fonológiában) megnőtt a teleológikus érvelés szerepe: az újraíró szabályok *célja* az, hogy valamilyen elveknek megfeleljen – vagy „jobban" megfeleljen – a nyelvtani alak. Az optimalitáselmélet ezeket a nyelvészeti trendeket formalizálja, és így a formális OT a *számítógépes elméleti nyelvészet* egyik legdinamikusabban fejlődő ága lett.

Hasonlóan a nyelvészeten kívüli – például fizikai, közgazdaságtani vagy pszichológiai – optimalizációs modellekhez, valamint közeli rokonához, a *harmónia-nyelvtanhoz* is [5], az OT különböző szempontokat (*constraints*, magyarul *megszorítások* vagy *korlátok*, vö. [6]) „gyúr össze" egyetlen célfüggvénnyé. Ezek a megszorítások gyakran egymással összeegyeztethetetlen és összemérhetetlen elvárásokat támasztanak a grammatikus alakkal szemben. A chomskyánus felfogással ellentétben, a grammatikus alakok megsérthetnek egyes megszorításokat, azonban a cél az, hogy „összességben minél jobban teljesítsenek".

Formálisan megfogalmazva: Egy u bemenetet (mögöttes alakot) a Gen *generátorfüggvény* a *jelöltek* (*candidates*: potenciális felszíni alakok) $\mathrm{Gen}(u)$ halmazára képezi le. Majd az optimalitáselmélet alapaxiómája azt mondja ki, hogy az u bemenethez tartozó $\mathrm{SF}(u)$ grammatikus felszíni alak optimalizálja a $H(c)$ célfüggvényt, a *Harmóniafüggvényt*:

$$\mathrm{SF}(u) = \underset{c \in \mathrm{Gen}(u)}{\arg\,\mathrm{opt}}\, H(c) \tag{1}$$

Az optimalitáselmélet a nyelvek (nyelvtípusok) közötti különbségeket eltérő célfüggvényekkel modellezi, melyeket más és más jelöltek optimalizálnak. Hogy az optimalizálás mit is jelent – maximalizálást vagy minimalizálást –, attól függ, hogy hogyan reprezentáljuk a célfüggvényt. Hagyományosan a $H(c)$ harmónia maximalizálásáról szokás beszélni. De az alábbiakban mi inkább megspórolunk magunknak egy negatív előjelet: a megszorítások sértéseinek a minimalizálása, és így a megszorításokból összerakott célfüggvény minimalizálása lesz a célunk.

Ha az egyes C_i megszorításokat a constraintek Con univerzális halmazából vett valós értékű függvényeknek tekintjük,[2] akkor ezek lineáris kombinációja egy valósértékű célfüggvényt eredményez:

$$H(c) = \sum_{i=0}^{n-1} g_i \cdot C_i(c) \tag{2}$$

Ezt nevezzük harmónianyelvtannak, és itt az (1)-beli optimum egyszerűen a valós számok halmazán vett minimumot jelenti. A lineáris kombináció g_i súlyai határozzák meg azt, hogy melyik megszorítás milyen eréllyel szól bele a grammatikus alak meghatározásába. A legtöbb nyelvészeten kívüli modell (például a közgazdaságtudományban és a kognitív tudományokban) hasonló optimalizációs elveket követ.

Ezzel ellentétben, az optimalitáselmélet nem valósértékű függvénnyé „gyúrja össze" a megszorításokat, hanem egy *hierarchiába* rangsorolja őket. A magasabbra rangsorolt megszorítás perdöntő: ha azt egy jelölt más jelölteknél súlyosabban sérti meg, akkor végképp elbukik, hiába viselkedik amúgy kitűnően az alacsonyabbra rendezett megszorítások szempontjából. Az ezen elvet (*szigorú dominancia*, *strict domination*) teljesítő harmóniafüggvényt többféle módon is reprezentálhatjuk: megszorításértések csomagjaként (multihalmazaként) [1], polinomokként vagy halmazelméleti rendszámokként [7]. A legegyszerűbb a vektorként történő reprezentáció, amelyeket lexikografikusan rendezhetünk az optimalizálás során:[3]

$$H(c) = \big(C_{n-1}(c), \ldots, C_1(c), C_0(c)\big) \tag{3}$$

A constraintek indexe tükrözi a rangsorolásukat: $C_{n-1} \gg \ldots \gg C_1 \gg C_0$. A c jelölthöz rendelt $H(c)$ vektor $n-i$-ik komponense a C_i megszorításnak felel meg, jelentése pedig az, hogy milyen mértékben (a legtöbb nyelvészeti modellben: hányszor) sérti meg a c jelölt a C_i megszorítást. A $H(c)$ vektor nem más, mint c sora az ismert OT-táblázatban, a csillagokat azok számával helyettesítve.

[2] Az optimalitáselmélet matematikailag helyes definíciójához azt is feltételeznünk kell, hogy az egyes megszorítások értékkészlete egy-egy jólrendezett halmaz [7]. A nyelvészeti gyakorlatban ez teljesül, hiszen a megszorítások általában nemnegatív egész értéket vesznek fel: nullát, ha a jelölt megfelel a megszorításbeli követelménynek, vagy egy pozitív egész számot, ha valahányszorosan megsérti azt.

[3] Lásd például [8]-t. [9, p. 1009] körbeírja a vektorreprezentációt, de nem nevezi néven. Tudtommal [10] hivatkozik először vektorokra, míg [11] a lexikografikus rendezésre. A két kifejezés [12]-ben találkozik először egymással.

Ha $H(c_1)$ lexikografikusan kisebb $H(c_2)$-nél, akkor c_1 harmonikusabb c_2-nél. Nevezzük *fatális megszorításnak* azt a C_f megszorítást, amelyre $C_f(c_1) \neq C_f(c_2)$, de minden magasabbra rendezett megszorítás azonosan értékeli ezt a két jelöltet. A fatális megszorítás felel meg a $H(c_1) - H(c_2)$ különbségvektor első nem-nulla elemének. Ez az elem határozza meg $H(c_1)$ és $H(c_2)$ lexikografikus rendezését: $C_f(c_1) < C_f(c_2)$ akkor és csak akkor, ha $H(c_1)$ lexikografikusan kisebb, mint $H(c_2)$. Átfogalmazva olyan formába, ahogy azt rövidesen használni fogjuk: ha c_1 harmonikusabb, mint c_2, akkor a fatális megszorítás c_1-et preferálja.

Mivel a H harmóniafüggvény értékkészlete n jólrendezett halmaz Descartes-szorzata, ezért maga az értékkészlet is jólrendezett halmaz a lexikografikus rendezés mellett. Következésképpen, valóban jól definiált az OT alapaxiómája:

$$\mathrm{SF}(u) = \underset{c \in \mathrm{Gen}(u)}{\arg\,\mathrm{opt}}\, H(c) \tag{4}$$

azaz az u bemenethez (mögöttes reprezentációhoz) tartozó $\mathrm{SF}(u)$ grammatikus felszíni reprezentáció optimalizálja a harmóniafüggvényt. Elvileg lehetséges, hogy két felszíni reprezentáció ugyanúgy sértse valamennyi megszorítást, és egyaránt optimalizálják a harmóniafüggvényt: ebben az extrém esetben az OT mindkét alakot grammatikusnak jósolja. A (4) egyenlőségben az optimalizálás lexikografikus minimalizálást jelent a fenti gondolatmenetünk értelmében. Azonban a szakirodalom, egy negatív előjelet helyezve $H(c)$ elé, a harmóniafüggvény maximalizálásáról beszél. E két megközelítés között nincs érdemi különbség.

Az optimalitáselmélet fősodra szerint mind a Gen függvény, mind a Con halmaz univerzális. A nyelvtanok közötti eltérést kizárólag a Con-beli megszorítások *rangsorolása* okozza. Két természetes nyelv nyelvtana a harmóniafüggvényükben különbözik egymástól, mégpedig abban, hogy a (3)-beli vektor komponenseit hogyan permutálják.

Optimalitáselméleti keretben a *tanuló algoritmus* feladata tehát a következő: adott (u_k, s_k) bemenet–kimenet párokhoz megtalálni azt a H függvényt, a komponensek azon permutációját, a megszorítások azon rangsorolását, amely mellett minden k-ra teljesül $s_k = \arg\,\mathrm{opt}_{c \in \mathrm{Gen}(u_k)} H(c)$. Az *offline algoritmusokban*, mint amilyen [13] *Recursive Constraint Demotion* algoritmusa, a tanítóadatokat, a mögöttes alak–felszíni alak párokat, egyszerre kapja meg a tanuló, mielőtt ezekből kikövetkeztetné a célnyelvtant. Ezek az algoritmusok azonban nyelvelsajátítási modellként kevéssé plauzibilisek. Így fordítsuk a figyelmünket inkább az *online algoritmusokra*, amelyek az adatokat folyamatosan adagolják a nyelvtanulónak.

Ez utóbbiak *hibavezérelt* (*error-driven*) megközelítések. A tanuló egy $H^{(0)}$ nyelvtannal (harmóniafüggvénnyel, megszorítás-rangsorolással) indul, amelyet fokozatosan módosít a megfigyelései függvényében. $H^{(0)}$ lehet egy véletlen hierarchia, vagy valamely „veleszületettnek” gondolt rangsorolás. Például a gyermeknyelvi adatok alapján szokás amellett érvelni, hogy kezdetben a jelöltségi (*markedness*) megszorítások magasabbra vannak rendezve, mint a hűségi (*faithfulness*) megszorítások. A tanulás egy pontján a tanuló által feltételezett $H^{(k-1)}$ nyelvtan predikciója az u_k-hoz tartozó jelöltre: $l = \arg\,\mathrm{opt}_{c \in \mathrm{Gen}(u_k)} H^{(k-1)}(c)$.

Ha ez az l (*loser form* a szakirodalomban) megegyezik a megfigyelt s_k-val (az alábbiakban w, mint *winner form*), akkor tanulónk örül a sikernek, és reménykedik, hogy elsajátította a célnyelvtant, minden más bemenetre is eltalálná a kimenetet. Amennyiben azonban l különbözik s_k-tól, a tanuló annak örül, hogy lehetősége van tanulásra: igyekszik úgy módosítani a nyelvtanát, hogy legközelebb $H^{(k)}$ már a helyes alakot jósolja. De legalábbis egy olyan nyelvtan felé közelítsen, amely a helyes w (azaz s_k) alakokat produkálja. A sikeres tanulás végén H^{∞} megegyezik a tanító H_t nyelvtanával, vagy legalább ekvivalens vele: minden (megfigyelhető) bemenetre azonos kimenetet jósol.

Hogyan módosítja a tanuló a nyelvtanát, amikor hibát észlel? Egyes megszorításokat feljebb, másokat lejjebb rangsorol annak érdekében, hogy közelebb kerüljön a célnyelvtanhoz. A tanító H_t nyelvtana, a célnyelvtan, az u_k mögöttes alakhoz a $w = s_k = \arg\,\mathrm{opt}_{c \in \mathrm{Gen}(u_k)} H_t(c)$ jelöltet rendeli. Mit jelent az, hogy l különbözik w-től? Azt, hogy H_t szerint w harmonikusabb l-nél, de $H^{(k-1)}$ szerint l harmonikusabb w-nél. Tehát, mint fentebb láttuk, a H_t-beli fatális megszorítás w-t kedveli, míg a $H^{(k-1)}$-beli fatális megszorítás l-t. A tanuló ebből azt a következtetést vonja le, hogy valamelyik w-t kedvelő megszorítást az l-t kedvelő megszorítások fölé kell rendeznie. Ezért az online OT tanulóalgoritmusok végigtekintik a Con-beli megszorításokat. Az l-t kedvelőket (vagy azok egy részét) lejjebb rendezik, a w-t kedvelőket pedig (esetleg) feljebb. Hogy pontosan hogyan teszik ezt, abban eltérnek egymástól a különböző algoritmusok [14,2,15,16,17,18].

3. Amikor a tanuló nem kap meg minden információt

Eddig feltételeztük, hogy a tanuló számára világos, melyik w jelölttel kell összevetnie az aktuális nyelvtana által generált l jelöltet. Ez azonban nincs mindig így, amint azt a bevezető fejezetben már láttuk. A megfigyelt nyelvi adat (*overt form*) nem feltétlenül jelölt OT értelemben (*candidate*). Utóbbi tartalmazhat olyan nyelvtani információt (például a szintaktikai frázisok és a fonológiai lábak határait jelző zárójeleket), amelyek az előbbiből hiányoznak. A hallható nyelvi adat nem feltétlenül felel meg egyetlen w jelöltnek, hanem jelöltek egy tágabb W halmazára képezhető csak le (például az azonos lineáris szerkezetet leíró fák erdejére). A W-beli jelöltek azonban egymástól eltérő módon sértik az egyes megszorításokat, és így a tanuló számára kérdéses marad, hogy mely megszorítást kell lejjebb, melyeket pedig feljebb rangsorolnia.

Egy korábbi kutatásban például a tagadó mondatok tipológiáját és történeti fejlődését vizsgáltuk [19]. A tagadószó (SN) megelőzheti az igét (SN V szórend, mint a magyarban, az olaszban és az ófranciában), követheti azt (V SN, mint a törökben vagy az élőnyelvi franciában), és körbe is veheti (SN V SN, mint az irodalmi franciában és az óangolban). Az utóbbi szórend azonban két különböző fastruktúrának is megfelelhet: [SN [V SN]] vagy [[SN V] SN]. A frázishatárok a szintaktikai elméleteknek szerves részei, de nem hallhatóak, nincsenek jelen a nyelvtanuló számára hozzáférhető nyelvi adatban. Az a nyelvtanuló gyermek, aki azt figyeli meg, hogy a célnyelv két részből álló tagadószerkezetet tartal-

maz (SN V SN), vajon miből fog rájönni, hogy a fenti két jelölt közül melyik grammatikus jövendőbeli anyanyelvében?

Tekintsük a következő (leegyszerűsített) példát. A Gen függvény a következő három jelöltet generálja (vagy a többi jelöltet már más megszorítások kiszűrték): [SN V], [[SN V] SN] és [SN [V SN]]. Három megszorításunk közül a *NEG minden egyes SN tagadószót egy megszorítássértéssel bünteti. A V-RIGHT és a V-LEFT megszorítások pedig a V-t közvetlenül tartalmazó frázis (mondjuk V' vagy VP) szerkezetére vonatkoznak: akkor teljesülnek, ha a V ennek a frázisnak a jobboldali, ill. baloldali eleme. Tehát a következő OT-táblázatot kapjuk:

	Tanuló →		← Tanító	
		*NEG	V-RIGHT	V-LEFT
l	[SN V]	1	0	1
w	[[SN V] SN]	2	0	1
	[SN [V SN]]	2	1	0

(5)

Képzeljük el, hogy a célnyelvtan V-LEFT ≫ V-RIGHT ≫ *NEG, vagyis a tanító (informáns) jobbról balra olvassa a fenti táblázatot. Számára az [SN [V SN]] alak a grammatikus, ami SN V SN-ként hangzik. Tegyük fel azt is, hogy a tanuló, pechjére, éppen az ellenkező hierarchiát feltételezi, a fenti táblázatot balról jobbra olvassa: *NEG ≫ V-RIGHT ≫ V-LEFT. Ő, ha rajta múlna, [SN V]-t mondana, de ez az l forma másként hangzik. Amint hallja a tanító által produkált alakot, észleli az eltérést, és beindul a hibavezérelt online tanuló algoritmusa. A nyelvtanát úgy szeretné módosítani, hogy SN V helyett legközelebb SN V SN-t mondjon. Azaz a nyelvtana egy másik jelöltet hozzon ki optimálisnak... Jó, de melyiket? [[SN V] SN]-t vagy [SN [V SN]]?

Tesar és Smolensky [14,2] azt javasolták, hogy a tanuló használja a saját nyelvtanát arra, hogy kiválassza az SN V SN két lehetséges értelmezése közül azt a w alakot, amellyel össze fogja vetni a saját maga által produkált l alakot. A tanuló nyelvtana felől (balról jobbra) nézve a táblázatot látjuk, hogy ő az [[SN V] SN] jelöltet jobbnak találja, mint az [SN [V SN]] jelöltet. Vagyis arra fog törekedni, hogy l helyett w-t hozza ki legközelebb optimálisnak. Több online OT tanulóalgoritmus létezik, amelyek részleteikben különböznek egymástól, de az alapgondolatuk azonos: ha egy megszorítás l-t jobbnak találja, mint w-t, akkor lejjebb kell rendezni (legalábbis, ha magasra volt eredetileg rangsorolva), ha pedig w-t találja jobbnak l-nél, akkor (bizonyos algoritmusban) feljebb.

Esetünkben egyetlen megszorítás van, amelyik eltérően értékeli l-t és w-t: a *NEG megszorítás l-t preferálja, és ezért lejjebb kell rangsorolni. A tanuló így eljuthat a V-RIGHT ≫ *NEG ≫ V-LEFT, majd a V-RIGHT ≫ V-LEFT ≫ *NEG hierarchiákhoz. Azonban, figyeljük meg, a tanuló mindvégig az [SN V] jelöltet fogja grammatikusnak tartani, a megfigyelt SN V SN alakot pedig mindig [[SN V] SN]-ként fogja értelmezni. Előbb-utóbb *NEG a rangsorolás aljára, a tanuló pedig patthelyzetbe kerül: az algoritmus elakad, az egyetlen átrangsorolandó megszorítást nincs már hova tovább átrangsorolni. A gondot az okozza, hogy a megoldás V-LEFT és V-RIGHT rangsorolásának a felcserélése lenne, de erre az algoritmus „nem jön rá". Mindvégig, amíg ez a csere nem történik meg, a

tanuló [SN V]-t tekinti l-nek és [[SN V] SN]-t w-nek, utóbbi produkálására törekszik. Ekkor valójában lehetetlent tűz ki célul: az [[SN V] SN] jelölt harmonikusan korlátolt (*harmonically bounded* [20]), egyetlen megszorítás szempontjából sem jobb, mint [SN V], és ezért nem létezik olyan rangsorolás, amely [[SN V] SN]-t hozná ki győztesnek. Hogyan lehet kitörni ebből a patthelyzetből?

Foglaljuk össze az eddigieket: a hibavezérelt online OT tanulóalgoritmusok (1) összehasonlítják a megfigyelt w jelöltet – vagy a megfigyelt alak egyik lehetséges w interpretációját – a tanuló által hibásan grammatikusnak vélt l jelölttel, és ha ezek egymástól eltérnek („hiba" lép fel), akkor (2) meghatározzák, hogy melyik megszorítás preferálja l-t, és melyik w-t, végül (3) előbbieket lejjebb, utóbbiakat feljebb rendezik. A *szétválasztás menetrendje*:

Minden $C_i \in \mathsf{Con}$ megszorításra,
1. ha $C_i(w) > C_i(l)$, akkor a C_i megszorítás l-t preferálja;
2. ha $C_i(w) < C_i(l)$, akkor a C_i megszorítás w-t preferálja.

Az l jelölt meghatározása, hibavezérelt algoritmusról lévén szó, természetesen a tanuló (egyelőre még) hibás nyelvtanától függ. A probléma abból származik, hogy szintén erre a hibás hierarchiára támaszkodunk w meghatározásánál, azaz a megfigyelés interpretálása során. Bár mindegyik W-beli jelölt ugyanúgy hangzik, de egyetlen w jelöltet választunk ki közülük a tanuló hibás nyelvtana segítségével. Egy rossz döntés ezen a ponton félreviheti az egész tanulási folyamatot. Milyen alapon bízzuk a tanító adatok értelmezését egy nyilvánvalóan téves hipotézisre? Tesar és Smolensky, amikor az eddigiekben leírt, *Robust Interpretive Parsing* (RIP, 'Robusztus Interpretatív Parszolás') nevű eljárásukat javasolták, az *Expectation–Maximization*-módszerek konvergenciáját látva azt remélték, hogy iteratív módon, előbb-utóbb, a tanuló eljuthat a célnyelvtanhoz. Sajnos azonban a kísérleteik azt mutatták, hogy ez nincs mindig így: néha végtelen ciklusba fut a tanuló, néha pedig – akárcsak a fenti példánkban – zsákutcába.

4. Két kiút a zsákutcából: Általánosított RIP

Figyeljük meg, hogy a szétválasztás fenti menetrendje során valójában érdektelen, hogy pontosan melyik jelöltet is választjuk w-nak. Ami számít, az w viselkedése az egyes megszorítások szempontjából. Nem szükséges rámutatnunk valamelyik jelöltre: elegendő meghatároznunk azt a határértéket, amellyel $C_i(l)$-t összehasonlítjuk. Ha $C_i(l)$ kevesebb a határértéknél, akkor a C_i megszorítás „l-et preferálja", és alacsonyabbra kell rangsorolni. Ha pedig $C_i(l)$ több, akkor C_i „w-t preferálja", és (az algoritmus részleteitől függően) magasabbra rangsorolandó. Az alábbiakban ezt a $C_i(W)$ határt az egész W halmazból számoljuk ki.

A fenti példánkban a tanuló, bár [SN V]-t mondana, de a hallott SN V SN alakról nem tudja eldönteni, hogy az hogyan interpretálandó: vajon a tanító nyelvtana szerint [[SN V] SN] vagy [SN [V SN]] a grammatikus? A maximumentrópia módszerek azt javasolják, ha nem tudunk dönteni két lehetőség közül, akkor adjunk mindkettőnek egyenlő esélyt. Tegyünk így most is, és átlagoljuk a táblázat két sorát:

		*Neg	V-right	V-left
l	[SN V]	1	0	1
w_1	[[SN V] SN]	2	0	1
w_2	[SN [V SN]]	2	1	0
W	w_1 és w_2 átlaga	2	0,5	0,5

(6)

A megfigyelt SN V SN alaknak potenciálisan két w felelhet meg. Ők alkotják a W halmazt. Az egyes megszorítások súlyozott átlaga értelmezhető ezen a W halmazon: valamely p_w súlyok mellett

$$C_i(W) = \sum_{w \in W} p_w \cdot C_i(w), \qquad \text{ahol} \qquad \sum_{w \in W} p_w = 1. \tag{7}$$

A (6) táblázatban a W halmaz mindkét elemére $p_w = 0,5$. Ha ezt az utolsó, átlagolt sort hasonlítjuk össze l sorával, arra a következtetésre jutunk, hogy *Neg mellett V-right is l-t preferálja, és mindkettőt lejjebb kell rangsorolni. Ráadásul V-left szempontjából pedig W a jobb, magasabban lenne a helye. Így tehát az algoritmus immár fel fogja tudni cserélni V-right és V-right rangsorolását. Vagyis a tanuló eljuthat a tanító nyelvtanához; de legalábbis egy azzal ekvivalens rangsoroláshoz, amelyben bár a megszorítások sorrendje eltérhet, de amely a célnyelvvel azonos nyelvet határoz meg.

A *szétválasztás menetrendje* a következőképpen módosul az ily módon bevezetett *Általánosított Robusztus Interpretatív Parszolás* nevű eljárásban [3]:

Minden $C_i \in \mathsf{Con}$ megszorításra, és valamely p_w értékek mellett,
1. ha $C_i(W) > C_i(l)$, akkor a C_i megszorítás l-t preferálja;
2. ha $C_i(W) < C_i(l)$, akkor a C_i megszorítás W-t preferálja.

Egyetlen kérdés maradt megválaszolatlanul: mi határozza meg a p_w értékeket a (7) képletben? Két közelmúltbeli cikkemben két különböző megoldást javasoltam. Egyiket a szimulált hőkezelés (szimulált lehűtés; *simulated annealing*), a másikat pedig a genetikai algoritmusok (*genetic algorithms*) ihlették.

4.1. GRIP: szimulált hőkezelés

A tanulás elején nem bízhatunk a tanuló nyelvtanában, mert az meglehetősen különbözhet a célnyelvtantól. Ha azonban hiszünk a tanulás sikerében, akkor fokozatosan növelhetjük a tanuló nyelvtanába vetett bizalmunkat. Ezért a tanulás elején a p_w súlyokat egyenlően szeretnénk elosztani W elemei között, a maximum-entrópia módszerek mintájára. A tanulás végén pedig oly módon, hogy csak a tanuló nyelvtana által legjobbnak tartott W-beli elem kapjon nullától különböző súlyt. Az utóbbi eset azonos a Tesar és Smolensky-féle eredeti RIP eljárással.

A *GRIP algoritmusnak* nevezett javaslatom [3] lényege az, hogy vezessünk be egy Boltzmann-eloszlást W-n. Ha $H(w)$ valós értékű, mint például a harmónia-nyelvtanban, akkor a Boltzmann-eloszlás alakja jól ismert:

$$p_w = \frac{e^{-H(w)/T}}{Z(T)}, \qquad \text{ahol} \qquad Z(T) = \sum_{w \in W} e^{-H(w)/T} \tag{8}$$

A termodinamikából kölcsönzött Boltzmann–Gibbs eloszlást egy pozitív T paraméter („hőmérséklet") jellemzi. Ha T nagyon magas ($T \gg H(w)$ minden $w \in W$-re), akkor a p_w súlyok (közel) egyenlően oszlanak el W elemei között. Ha viszont T nagyon alacsony ($0 < T \ll H(w)$), akkor a súly nagy része a legalacsonyabb $H(w)$ „energiájú" elem(ek)re koncentrálódik. Az optimálistól eltérő W-beli elemek p_w értékei nullához tartanak. A *szimulált hőkezelés* (szimulált lehűtés) név alatt ismert eljárások lényege az, hogy az algoritmus T paramétere nagyon magas értékről nagyon alacsony értékre fokozatosan csökken le.

A szimulált hőkezelés optimalizációs eljárásként ismert, és korábban ekként alkalmaztam az OT-ban is. Az *SA-OT algoritmus* egy performancia-modell: egy heurisztikus módszer az optimális jelölt megkeresésére [21,8,7]. Most azonban nem az optimális jelöltet keressük, hanem nyelvtant tanulunk.

Az *Általánosított Robusztus Interpretatív Parszolás* eljárás újítása az, hogy nem egyetlen w viselkedését veti össze az l viselkedésével megszorításonként, hanem az összes lehetséges W-beli jelölt viselkedésének súlyozott átlagát. A p_w súlyokat kell tehát meghatároznunk, és *erre* használjuk a Boltzmann-eloszlás (8) képletét. Arra tehát, hogy az egyes megszorítások W-n vett súlyozott átlagát definiáló (7) képletben szereplő p_w súlyokat kiszámítsuk. Majd, a tanulás során fokozatosan csökkentjük a (8)-ban használt T értékét, és ezáltal módosulnak a súlyok is. Kezdetben W minden eleme hozzájárul a megszorítások átrangsorolásának meghatározásához. Később azonban csak azok a jelöltek, amelyek a tanuló nyelvtana szerint a legharmonikusabbak W-ben.

Az algoritmusból azonban egy csavar még hiányzik. A (8) képlet valósértékű $H(w)$ függvényt feltételez. De az optimalitáselméletben $H(w)$ vektorértékű, amint azt (3) alatt láttuk. Ezért az idézett cikkemben a (8) Boltzmann-eloszlást vektorértékű $H(w)$-ra is értelmeznem kellett. Az eredmény formailag sok szempontból hasonlít az SA-OT algoritmusra. A Boltzmann-eloszlás T „hőmérséklet" paraméterének szerepét egy (K, t) paraméterpár veszi át, és ezek határozzák meg a p_w súlyokat. Az eljárás mögött húzódó matematikai gondolatmenet, valamint a pszeudokód és annak elemzése megtalálható [3]-ben – itt hely hiányában nem térhetünk ki ezekre a részletekre.

Ha a (K, t) paraméter már a tanulási folyamat elején is nagyon alacsony, akkor visszajutunk a hagyományos RIP eljáráshoz. Vajon a GRIP algoritmussal, magasabb (K, t) kezdőértékek mellett, javítható a tanulás sikeressége?

4.2. JRIP: „genetikai algoritmus"

[4] egy másik – matematikailag egyszerűbb – megközelítést mutat be a p_w súlyok meghatározására. Az alfejezet címében szereplő idézőjelek arra utalnak, hogy az alábbiakban leírtak csak távolról emlékeztetnek a genetikai algoritmusokra: nincs mutáció és szelekció, csupán egy változó összetételű rangsorolás-populáció, amely, remélhetőleg, konvergál a „megoldás" felé.

Yang [22] gondolatát követve, a javaslat lényege az, hogy a tanuló nem egy, hanem r darab nyelvtannal (esetünkben megszorítás-rangsorolással) rendelkezik. Ezeket külön-külön, véletlenszerűen inicializáljuk, és külön-külön tanulnak a RIP algoritmus szerint. A k-ik hierarchia ($1 \leq k \leq r$) minden egyes bejövő adat után kiszámítja a maga l_k és w_k jelöltjeit: ő maga mely jelöltet választaná, illetve a megfigyelt alak mely interpretációját találja optimálisnak. Ha ezek után a k-ik hierarchia összehasonlítja l_k-t w_k-val, lejjebb sorolja az l_k-t preferáló megszorításokat, és feljebb sorolja a w_k-t kedvelőket, akkor visszajutunk a hagyományos RIP algoritmushoz. Ha nem is mindegyik nyelvtan, de valamelyik közülük előbb-utóbb a célnyelvtanhoz fog konvergálni.

Ez a megközelítés azonban nem lenne plauzibilis gyermeknyelv-elsajátítási modell. Mind a k hierarchia csak kis valószínűséggel fog egyszerre sikerrel járni [4]. Ha pedig a nyelvtanok egy része nem jut el a célnyelvtanhoz, akkor a felnőttek honnan tudják, hogy melyik nyelvtant kell használniuk? A teljes nyelven tesztelik valamennyi nyelvtant? Számítógépes kísérletek játéknyelvtanai esetén egy ilyen teszt még elképzelhető lenne, de nem valódi nyelv esetén.

Ezért javasolom, hogy az egyes hiararchiák a saját maguk által optimálisnak tartott l_k jelöltet ne a saját maguk által meghatározott w_k jelölthöz hasonlítsák, hanem valamennyi w_k „átlagához". A rangsorolások a *hierarchiák populációjában* közösen interpretálják a bejövő alakot, hátha közös erővel sikeresebbek, mint egyenként. Közösen határozzák meg azt a $C_i(W)$ határértéket, amellyel utána mindenki külön-külön összeveti a saját $C_i(l_k)$-ját, hogy eldöntse, lejjebb vagy feljebb rangsorolja-e a C_i megszorítást a saját hierarchiájában. Sikeres tanulás esetén mind az r rangsor a célnyelvtanhoz konvergál.

Így jutunk el a *JRIP algoritmushoz*. A (7) képlet a következő alakot veszi fel:

$$C_i(W) = \frac{1}{r} \sum_{k=1}^{r} C_i(w_k) \tag{9}$$

Másképp megfogalmazva, a (7) egyenletbeli p_w arányos azon populációbeli nyelvtanok számával, amelyek w-t választották w_k gyanánt a W halmazból.

Az $r = 1$ eset megfelel a hagyományos RIP algoritmusnak. Vajon növelhető a tanulás sikere JRIP-pel, ha magasabb r-t választunk?

5. Szóhangsúly

A tagadó mondat eddig tárgyalt szórendjéhez hasonló problémával szembesül a tanuló (algoritmus) a hangsúly elsajátításánál is. A szóhangsúly kurrens fonológiai elméletei a szótagokat *lábakba* szervezik, de ezek nem „hallhatóak". Következésképp a tanuló nem tudhatja, hogy például a *hókusz-pòkusz* négyszótagú szó jambikus vagy trochaikus nyelvre bizonyíték-e. Elemezhető akár [*hók*][*uszpòk*]*usz*-ként, akár [*hókusz*][*pòkusz*]-ként. A szóhangsúly példáján mutatta be [2] a RIP algoritmust, és ezért én is ezen a példán illusztálom, hogy az általam javasolt két új módszer mennyit képes javítani a RIP algoritmuson.

A metrikus fonológia szerint a szótagok metrikus lábakba szerveződhetnek. Egy láb egy vagy két szótagból állhat. Az egyik láb kiemelt: a „feje" kapja a szó

főhangsúlyát. A többi láb feje mellékhangsúlyt kap. A két szótagból álló lábak másik szótagja, valamint a lábakon kívül eső szótagok nem kapnak hangsúlyt. A metrikus fonológia OT modelljeiben a megszorítások vonatkozhatnak a szótagokra (például nehéz szótag kapjon hangsúlyt; ne kerüljön szótag a lábakon kívülre), a lábakra (például a láb legyen kétszótagú; a láb legyen jambikus) és az egész szó szerkezetére (például a szó bal határa essen egybe egy láb bal határával). Kísérleteim során ugyanazt az OT metrikus fonológiai szakirodalomban széles körben elterjedt tizenkét megszorítást használtam, mint Tesar és Smolensky [2].

A kísérlet elején mind a tanító, mind a tanuló nyelvtanát véletlenszerűen inicializáltam. A tizenkét megszorításhoz egy-egy 0 és 50 közötti lebegőpontos rangsorértéket rendeltem, Boersma és Magri algoritmusainak megfelelően [16,18], eltérően az eredeti *EDCD* algoritmustól [14,2]. Minél magasabb egy megszorítás rangsorértéke, annál magasabbra kerül a hierarchiában. Négy algoritmust vizsgáltam: Boersma *GLA*-je az l-t preferáló megszorítások rangsorértékét 1-gyel csökkenti, és a W-t preferáló megszorításokét 1-gyel növeli. Magri algoritmusa a legmagasabbra rangsorolt, l-t preferáló megszorítás rangsorértékét 1-gyel csökkenti, és az összes – n darab – W-t preferáló megszorításét $1/n$-nel növeli. Az Alldem algoritmus csak az l-t preferáló megszorításokhoz nyúl, míg a Topdem algoritmus kizárólag a legmagasabbra rangsorolt, l-t preferáló megszorítás rangsorértékét csökkenti (szintén 1-gyel).

A nyelvtanuló feladata egy négy szóból álló lexikon helyes hangsúlyozásának a megtanulása volt. A lexikon szavai négy és öt, könnyű és nehéz szótagokból álltak: *ab.ra.ka.dab.ra*, *a.bra.ka.da.bra*, *ho.kusz.po.kusz* és *hok.kusz.pok.kusz*. A tanító ezeket látta el szóhangsúllyal a saját nyelvtana szerint, majd törölte a lábhatárokat, és az így generált nyelvi adatokat ismételgette a tanulónak. A tanulás akkor volt sikeres, ha a tanuló talált olyan hierarchiát, amellyel reprodukálta az általa megfigyelt nyelvi adatokat. Egy-egy paraméterbeállítás mellett a kísérletet több ezerszer megismételtem, és mértem a sikeres tanulások arányát.

Amikor a GRIP és a JRIP paraméterei a hagyományos RIP-nek feleltek meg, a sikeres tanulás aránya 76-78% körül volt, az algoritmus részleteitől függően. Megfelelő paraméterbeállításokkal azonban ez az arány jóval 90% fölé – néhány további trükkel pedig akár 95% fölé is – emelkedett [3,4]. A különbség statisztikailag erősen szignifikáns, bizonyítván a GRIP és JRIP algoritmusok sikerét.

6. Összefoglalás és utószó

Bemutattam, hogy az OT tanulóalgoritmusok milyen problémával szembesülnek, ha a tanítóadatok nem tartalmaznak minden fontos információt. A megfigyelhető adat lehetséges értelmezései közül a hagyományos RIP eljárás a tanuló nyelvtana szempontjából legjobbat választja. Ehelyett az értelmezések megszorításssértései átlagolását javasoltam, két különböző módszerrel. A szóhangsúllyal folytatott kísérletek során mindkét módszer szignifikánsan javított a RIP hatékonyságán.

A konferenciaabsztrakt megírása óta eltelt két hónap. Kislányom időközben elsajátította az /e/ és az /i/ közötti fonemikus különbséget a magyar nyelv nyelvtanában. Vajon milyen tanulóalgoritmust használt?

Hivatkozások

1. Prince, A., Smolensky, P.: Optimality Theory: Constraint Interaction in Generative Grammar. Blackwell, Malden. Eredetileg: *Technical Report nr. 2. of the Rutgers University Center for Cognitive Science* (RuCCS-TR-2) (1993/2004)
2. Tesar, B., Smolensky, P.: Learnability in Optimality Theory. MIT Press, Cambridge, MA – London (2000)
3. Biró, T.: Towards a Robuster Interpretive Parsing: Learning from overt forms in Optimality Theory. Journal of Logic, Language and Information (accepted)
4. Biró, T.: Uncovering information hand in hand: Joint Robust Interpretive Parsing in Optimality Theory. Linguistic Inquiry (submitted)
5. Smolensky, P., Legendre, G., eds.: The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar. MIT Press (2006)
6. Rebrus, P.: Optimalitáselmélet. In Siptár, P., ed.: Szabálytalan fonológia. Tinta Könyvkiadó, Budapest (2001) 77–116
7. Bíró, T.: Finding the Right Words: Implementing Optimality Theory with Simulated Annealing. PhD thesis, University of Groningen (2006) ROA-896.
8. Bíró, T.: A sz.ot.ag: Optimalitáselmélet szimulált hőkezeléssel. In Alexin, Z., Csendes, D., eds.: III. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, SzTE Informatikai Tanszékcsoport (2005) 29–40
9. Ellison, T.M.: Phonological derivation in Optimality Theory. In: Proceedings of the 15th CoLing Conference. Volume 2. (1994) 1007–1013
10. Eisner, J.: Efficient generation in primitive Optimality Theory. In: Proceedings of the 8th conference of EACL. (1997) 313–320
11. Tesar, B., Grimshaw, J., Prince, A.: Linguistic and cognitive explanation in Optimality Theory. In Lepore, E., Pylyshyn, Z., eds.: What is Cognitive Science? Blackwell, Malden, MA (1999) 295–326
12. Eisner, J.: Easy and hard constraint ranking in Optimality Theory: Algorithms and complexity. In Eisner, J., Karttunen, L., Thériault, A., eds.: Finite-State Phonology: Proc. of the 5th SIGPHON Workshop, Luxembourg (2000) 57–67
13. Tesar, B.: Computational Optimality Theory. PhD thesis, University of Colorado, Boulder (1995) ROA-90.
14. Tesar, B., Smolensky, P.: Learnability in Optimality Theory. Linguistic Inquiry **29**(2) (1998) 229–268
15. Boersma, P.: How we learn variation, optionality, and probability. Proceedings of the Institute of Phonetic Sciences, Amsterdam (IFA) **21** (1997) 43–58
16. Boersma, P., Hayes, B.: Empirical tests of the Gradual Learning Algorithm. Linguistic Inquiry **32** (2001) 45–86 ROA-348.
17. Boersma, P.: Some correct error-driven versions of the Constraint Demotion algorithm. Linguistic Inquiry **40**(4) (2009) 667–686
18. Magri, G.: Convergence of error-driven ranking algorithms. Phonology **29**(2) (2012) 213–269
19. Lopopolo, A., Biró, T.: Language evolution and SA-OT: The case of sentential negation. Computational Linguistics in the Netherlands J **1** (2011) 21–40
20. Samek-Lodovici, V., Prince, A.: Optima. ROA-363 (1999)
21. Bíró, T.: How to define Simulated Annealing for Optimality Theory? In: Proceedings of Formal Grammar/Mathematics of Language, Edinburgh (2005)
22. Yang, C.D.: Knowledge and Learning in Natural Language. Oxford University Press, Oxford, UK (2002)